msmsEDA and msmsTests: R/Bioconductor packages for spectral count label-free proteomics data analysis

Josep Gregori, Alex Sanchez, and Josep Villanueva
Vall Hebron Institute of Oncology &
Statistics Dept. Barcelona University
josep.gregori@gmail.com

May 21, 2014

Abstract

Background
The study of the secretome of cancer cell-lines by label-free LC-MS/MS simplifies the problem of finding potential biomarkers because the much restricted dynamic range of the proteins compared to plasma, and because secretomes may be a surrogate of proximal fluids enriched with disease specific proteins. Dealing with counts requires of adequate statistical tools for inference, with linear models able to account for covariates in a distribution specific context. On the other hand label-free experiments require of good experimental design and of tools for the detection and correction of batch effects due to non controlled factors. Also because reproducibility is of the biggest importance in biomarker discovery, besides significance, biological relevance is required, so that declared differentially expressed proteins could not rely solely in p-values, and should show a minimum signal and an minimum effect size.

The software
Tools and solutions to the challenges described above are provided by the two R/Bioconductor packages `msmsEDA` and `msmsTests`. `msmsEDA` is a package providing a set of functions for the exploratory data analysis (EDA) of label-free shotgun proteomics datasets, devised to detect outliers and confounding factors. `msmsTests` is a package with functions to test differential expression between two biological conditions. The tests are based on a GLM model with offsets as normalizing factors, and eventual blocking factors. The Poisson and the negative binomial distributions, or the quasi-likelihood are considered. The package includes additional functions for the interpretation of results, and for post-test filtering

Availability
Both packages are freely available at http://www.bioconductor.com

Keywords biomarker discovery, proteomics, spectral counts, LC-MS/MS, EDA, batch effects, GLM, Poisson, binomial negative, quasi-likelihood, R package, Bioconductor, MSnbase, edgeR, msmsEDA, msmsTests

Contents

1 Background

1.1 Proteomics in biomarker discovery

While DNA contains full information on the plans of a cell, and mRNA works as a messenger with pieces of information send to the machinery of the cell, proteins are the functional part and more accurately reflect the phenotype. So, as the protein lays in a higher functional level than mRNA, it is expected that proteomics could bring to the discovery of molecular targets and biomarkers more effectively that transcriptomics did [Gygi et al., 1999]. The immediate target would be to study the proteome of tumor cells, nevertheless a very large fraction of the protein lysate corresponds to structural proteins, like cellular organelles, proteosome and proteins related to cellular core functions and protein translation. The fraction of disease specific proteins in the whole cell lysate is negligible [Tirumalai et al., 2003]. Looking for non-invasive tests the best source would be blood plasma, the most comprehensive human proteome containing proteins from all tissues and processes, with disease specific secreted proteins. But the proteome in plasma shows big complexity and a high dynamic range. With typical proteins like albumin accounting for over 50% of the total of proteome, and the top 22 proteins giving the 99% of the protein content of plasma, the relative concentration of disease-specific

biomarkers is expected to be very low except in fortuitous cases [Tirumalai et al., 2003]. As biomarkers specific for a particular disease arise locally from the affected tissue, it is expected that a fluid closer or in direct contact with the tissue will be enriched in these highly informative molecules. These proximal fluids are local sinks for proteins secreted or leaked from the tissue. Of particular interest is the interstitial fluid [Rifai et al., 2006]. The use of the cancer secretome has recently been proposed to interrogate tissue-proximal fluids and conditioned media of cell lines for biomarker discovery (BD) [Stastna & Van Eyk, 2012]. The presence of growth factors and proteases in these fluids, indicates that secretomes might help in monitoring critical aspects of cancer progression such as invasion and metastasis. In fact, a significant fraction of abnormally regulated genes in cancer encode secreted proteins [Gronborg et al., 2006; Lawlor et al., 2009; Mathias et al., 2009].

1.2 Quantitation by spectral counts in LC-MS/MS

Protein quantification by LC-MS/MS may be measured by chromatogram ion intensities [Sandin et al., 2011] or by the number of spectral counts (SpC) [Lundgren et al., 2010] assigned to peptides belonging to a protein. The software introduced in this paper deals with SpC quantification. A recent expert review [Lundgren et al., 2010] considers instrument aspects, inherent limitations of SpC, common normalizations by protein length or mass, and relative or absolute quantification by SpC.

1.3 Label-free proteomics

In biomarker discovery we seek for differential expression between two biological conditions, generally disease and control. That is, unbiased relative quantification. To ensure this unbiased comparison, labelled procedures have been employed both in transcriptomics and in proteomics. Labelled procedures allow the early mix of the samples to be compared. Since they are pooled they are processed together and any non controlled factor affects equally both conditions. The labels allow the posterior identification of features belonging to each condition. Different approaches have been used. Among the most popular [Patel et al., 2009]: Stable isotope labelling by amino acids in cell culture (SILAC), isotope-coded affinity tags (ICAT), and isobaric tags for relative and absolute quantification (iTRAQ). The later is commercially available with eight isobaric tags. Despite the advantage of labelled approaches they have potential limitations. Complex preparation steps, requirements for increased sample concentration and incomplete labelling, are the main issues. Labelling usually requires fractionation since all samples are analyzed togheter, causing a drop in sensitivity. The key of this approach is that all samples to be compared are processed together. In this way the experiments are fully dedicated to a unique comparison and the acquired data, in most cases, may not be reused with different purposes. Label-free proteomic analysis provide a more flexible alternative. This means that each sample is processed and analysed separately. The counterpart is high risk of bias due to non controlled factors affecting differently one condition than the other [Neilson et al., 2011; Sandin et al., 2011; Zhu et al., 2010; Patel et al., 2009]. This requires carefully planned and designed experiments, to minimize confounding and bias as much as possible. Nevertheless, given the sample sizes attainable not all factors

potentially contribuiting to the results may be controlled. And this brings almost unavoidably to the expression of batch effects in the label-free LC-MS/MS dataset, mainly when the period of collecting and analysing the different samples spans a long period of time [Gregori et al., 2012].

1.4 Batch effects

In words of T. Speed in the foreword to [Scherer, 2009] *"Samples might come in one at a time, over months or years, but are commonly collected in batches. However the collection, processing and analysis are conducted, time or batch effects are unavoidable. Important though design is, and it is rightly emphasized in the book, there is in general little chance of entirely eliminating these effects. We must do our best with good design, but we must also plan to be in a position to identify and subsequently correct for those effects we are unable to eliminate by design."*

The batch effect is defined to be systematic and unintentional, in contrast with the experimental noise which is random in nature. It refers exclusively to systematic technical differences when samples are processed and measured in different batches or in different times. Because of its systematic nature, the worst manifestation of batch effects is bias. These effects may be visualized by multidimensional techniques like Principal Components Analysis (PCA), Singular Value Decomposition (SVD), Hierarchical Clustering (HC), or Heatmaps (HM) [Scherer, 2009; Luo et al., 2010; Gregori et al., 2012]. Ideally the samples should cluster by treatment level, with independence of the time in which they were treated and measured. Once identified there a number of proposed methods to correct for these batch effects [Scherer, 2009; Luo et al., 2010; Lazar et al., 2013].

1.5 Reproducibility

In biomarker discovery reproducibility is of the maximum concern. As evidenced by the MAQC-1 study [Shi et al., 2006] the main reason for the lack of reproducibility of lists of differentially expressed genes found with oligonucleotide microarray (MA) experiments was relying solely on p-values [Shi et al., 2008]. The recommendation was to rank the lists of differentially expressed genes by fold change instead of by p-value, giving relevance to the effect size beyond the significance. This may be extended by saying that significant features with low signal will be scarcely reproducible because in different samples may scape detection, both because of the sampling process and because of technical reasons related to the noise level. Significant features with a low effect size will be scarcely reproducible because they will obtain a significant p-value only when the observed variance is low enough. Significant features with low signal and low effect size will show very bad reproducibility. The combination of good signal and effect size is the best guarantee for reproducibility. This has also been evidenced with SpC label-free data [Gregori et al., 2013].

1.6 Cell-to-cell normalization

Differential expression in the omics is based on some basic assumptions, not always explicitly given. When these assumptions are roughly fulfilled the comparisons between

two biological states may be considered as nearly unbiased, provided that a good experimental design is used [Knudsen, Knudsen]. These assumptions are usually taken for granted and receive no criticism in most, if not all, studies. In transcriptomics and proteomics, where equal amounts of substance gathered from two biological conditions are measured, it is considered that the cells produce globally almost equal quantity of total substance. Under this assumption comparing equal amounts of substance corresponds to comparing the substance produced by equal number of cells. And this is a cell to cell comparison, where the cell is the biological unit of interest. When studying secretomes this assumption deserves a careful consideration, as the number of involved proteins is drastically reduced, and no structural or related to the metabolism proteins are expected. One biological state may be globally stimulated or depressed, in secretome terms, respect to the other, and hence the basic assumption may fail, giving rise to biased comparisons if not proper data normalization is used.

1.7 Packages `msmsEDA` and `msmsTests`

We developed two R/Bioconductor [R Core Team, 2013; Gentleman et al., 2004] packages which offer solutions to the discussed challenges in SpC label-free projects. Although optimised to work with secretomes, they are equally useful in the analysis of any SpC label-free project. `msmsEDA` provides functions and tools to assess the quality of a LC-MS/MS project, and to detect and visualise outliers or batch effects. `msmsTests` includes functions to test the differential expression of proteins in SpC matrices by GLM, using either the Poisson or the negative binomial (NB) distribution, or the quasi-likelihood (QL) extension. These functions take the formulas of the two models to be compared as parameters, and allow for blocking factors to correct observed batch effects. The NB method in `msmsTests` uses the implementation in the package `edgeR` [Robinson & Smyth, 2008] wich is a bayesian approach to share information across proteins and allows the estimation of the dispersion with fewer replicates. Flexible normalization factors are incorporated into the GLM model by means of offsets. Another function flags the proteins as significant and likely reproducible, based on multi-test adjusted p-values, signal strength and effect size. Both packages provide utility functions to help in the interpretation of the results.

2 Results

To illustrate the use of these packages we use an spikings dataset with 500ng of yeast lysate spiked with and 200fm and 600fm of a mix of 48 equimolar human proteins (UPS1). Then to show the need for the cell-to-cell normalization in secretome analysis we give the observed secretome production in pg/cell of different cancer cell-lines at base line, and under different treatments.

2.1 Model, batch effects and post-test filters

The spikings dataset consists of four technical replicates of each condition measured the same day, and three extra technical replicates of each condition measured a few weeks

later. It consists then of seven replicates of each condition measured in two batches. As all are technical replicates of the same digest we would not expect a significant contribution of non controlled factors, nevertheless an EDA shows bigger similarity among samples of the same batch than among samples of the same condition (See Fig. 1). This indicates that a model with a blocking factor, where its levels are the two batches, may improve the power of the tests in BD.

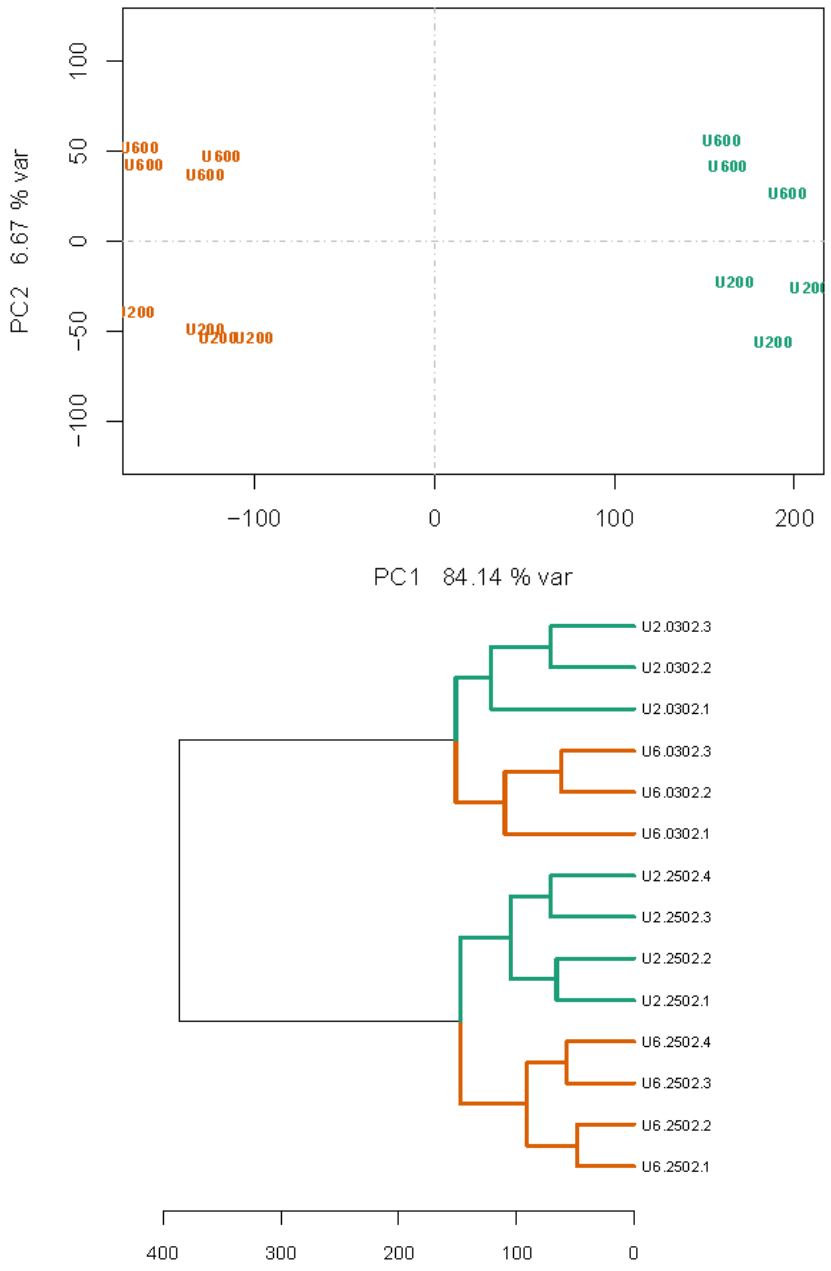

Figure 1: Above: Hierarchical Clustering of the samples. Below: Scaterplot on the two principal components of the PCA. Batch effects are evidenced when the samples cluster by processing date instead of by biological condition. The two biological conditions are U200 and U600, the two processing dates are 0302 and 2502

On the other hand, we wish to obtain a list of differentially expressed proteins (DEP) highly reproducible. Improving the reproducibility means increasing the confidence to declare the same DEPs in a new experiment, possibly made in a different laboratory and/or platform. If we are concerned by the reproducibility of our list of DEPs, we may increase the stringency of the p-value threshold. This will likely reduce the false positives though also the true positives. Another possibility, as explained above, is to filter out the DEPs having low signal and/or low effect size.

We explored the results using the three GLM methods implemented in `msmsTests` under six different settings: i) p-value threshold of 0.05. ii) increasing the stringency of the p-value to 0.01 to increase the reproducibility. iii) using a post-test filter to exclude the DEGs of poor reproducibility, as those with less than 2 SpC in the most abundant condition, or with an absolute logFC less than 0.8. iv) the same as (i) but with a model with a blocking factor for the batches. v) as (ii) with a blocking factor. And vi) as (iii) with a blocking factor.

Table 1 shows the results in terms of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Four test scores are given. The true predictive rate (TPR) giving the sensitivity, the accuracy (ACC) as the fraction of observed trues, the false discovery rate (FDR) as the fraction of FP respect to the positives, and its complement, the positive predictive value(PPV) as the fraction of TP respect to the positives. The R scripts used to obtain the Figure 1 and the results in Tables 1 and 2, using tools in the two packages are given as supplementary material.

The GLM with the Poisson distribution is almost insensitive to the inclusion of a blocking factor, as it is not affected by the observed variance of the data. Both the p-value and the estimated logFC are only slightly affected. The post-test filter gives the higher PPV, with no FP.

The quasi-likelihood GLM shows a big improvement in TP when using the model with a blocking factor, although at the cost of a high number of FP. This behaviour is explained by the reduction in the residual variance when using a blocking factor in the model, and by the fact that the QL is not robust to very low observed variances [Leitch et al., 2012]. The use of a more stringent p-value cut-off does not solve the situation. The post-test filter gives the higher accuracy and PPV, eliminating all FPs. The QL benefits specially of the post-test filter.

The NB shows a behaviour similar to the Poisson, although the use of a blocking factor gives improvements in the TPR. This may be explained by the fact that the NB brings to the Poisson when the dispersion parameter is 0 [Agresti, 2002], and because no overdispersion is expected when dealing with technical replicates. In a setting with expected biological variability the NB should give better results than the Poisson [Agresti, 2002; Robinson & Smyth, 2008].

The behaviour of the three tests is further characterized by the results on a single batch, with only three technical replicates of each condition, collected in Table 2. i) As expected, a smaller sample size brings to the identification of less proteins and less positives. ii) The results of the Poisson and of the NB are undistinguishable. iii) The QL

Table 1: Truth tables for different models and significances, with true predictive rate (TPR = TP/(TP+FN)), accuracy (ACC=(TP+TN)/(TP+TN+FP+FN)), false discovery rate (FDR=FP/(FP+TP)) and positive predictive value (PPV=TP/(TP+FP)). The models are a) Treat: $y \sim Treatment$ vs $y \sim 1$, b) Treat+Fltr: $y \sim Treatment$ vs $y \sim 1$ with post test filter, c) Treat+Batch: $y \sim Treatment + Batch$ vs $y \sim Batch$, d) Treat+Batch+Fltr: $y \sim Treatment + Batch$ vs $y \sim Batch$ with post test filter.

Model	Signif	Distr	TP	FP	TN	FN	TPR	ACC	FDR	PPV
Treat	0.05	Pois	37	2	627	9	80.43	98.37	5.13	94.87
Treat	0.01	Pois	33	1	628	13	71.74	97.93	2.94	97.06
Treat+Fltr	0.05	Pois	37	0	629	9	80.43	98.67	0.00	100.00
Treat+Batch	0.05	Pois	37	2	627	9	80.43	98.37	5.13	94.87
Treat+Batch	0.01	Pois	33	1	628	13	71.74	97.93	2.94	97.06
Treat+Batch+Fltr	0.05	Pois	37	0	629	9	80.43	98.67	0.00	100.00
Treat	0.05	QL	32	1	628	14	69.57	97.78	3.03	96.97
Treat	0.01	QL	22	0	629	24	47.83	96.44	0.00	100.00
Treat+Fltr	0.05	QL	31	0	629	15	67.39	97.78	0.00	100.00
Treat+Batch	0.05	QL	41	18	611	5	89.13	96.59	30.51	69.49
Treat+Batch	0.01	QL	36	13	616	10	78.26	96.59	26.53	73.47
Treat+Batch+Fltr	0.05	QL	38	0	629	8	82.61	98.81	0.00	100.00
Treat	0.05	NB	35	0	629	11	76.09	98.37	0.00	100.00
Treat	0.01	NB	32	0	629	14	69.57	97.93	0.00	100.00
Treat+Fltr	0.05	NB	35	0	629	11	76.09	98.37	0.00	100.00
Treat+Batch	0.05	NB	37	2	627	9	80.43	98.37	5.13	94.87
Treat+Batch	0.01	NB	33	1	628	13	71.74	97.93	2.94	97.06
Treat+Batch+Fltr	0.05	NB	37	0	629	9	80.43	98.67	0.00	100.00

shows poor results with just three replicates. And iv) The post-test filter gives far better results, in terms of restricting the FP without compromising the TP, than increasing the p-value stringency in all three cases. A previous work [Gregori et al., 2013] with extensive simulations confirmed the post-test filter as the best option for reproducibility.

Each experimental settings will require a corresponding model. With no replicates or with just two technical replicates, the GLM Poisson is the best option. With two or three biological replicates of each condition, some overdispersion may be expected, and the negative binomial implemented by edgeR [Robinson & Smyth, 2008] is the best option. With more that four replicates both the quasi-likelihood or the negative binomial may be appropriate.

The Poisson model will not benefit of a blocking factor when batch effects are detected, but the QL and the NB are expected to show higher power. The post-test filter will be a better option to improve the reproducibility than increasing the p-value stringency, specially with the QL.

Table 2: Truth tables with true predictive rate (TPR = TP/(TP+FN)), accuracy (ACC=(TP+TN)/(TP+TN+FP+FN)), false discovery rate (FDR=FP/(FP+TP)) and positive predictive value (PPV=TP/(TP+FP)). The models are a) Treat: $y \sim Treatment$ vs $y \sim 1$, b) Treat+Fltr: $y \sim Treatment$ vs $y \sim 1$ with post test filter.

Model	Signif	Distr	TP	FP	TN	FN	TPR	ACC	FDR	PPV
Treat	0.05	Pois	29	0	565	14	67.44	97.70	0.00	100.00
Treat	0.01	Pois	23	0	565	20	53.49	96.71	0.00	100.00
Treat+Fltr	0.05	Pois	29	0	565	14	67.44	97.70	0.00	100.00
Treat	0.05	QL	20	4	561	23	46.51	95.56	16.67	83.33
Treat	0.01	QL	6	2	563	37	13.95	93.59	25.00	75.00
Treat+Fltr	0.05	QL	20	0	565	23	46.51	96.22	0.00	100.00
Treat	0.05	NB	29	0	565	14	67.44	97.70	0.00	100.00
Treat	0.01	NB	23	0	565	20	53.49	96.71	0.00	100.00
Treat+Fltr	0.05	NB	29	0	565	14	67.44	97.70	0.00	100.00

2.2 Cell-to-cell normalization

The flexible use of offsets as normalizing factors in GLM models allows an easy implementation of the cell-to-cell normalization in cell-line secretomes as described under methods by equation 11. When observing the secretome production of different cancer cell lines at base line, or of different biological conditions for a single cell line, as given in Table 3, it is clear that an unbiased cell-to-cell comparison of secretomes requires this sort of normalization.

Table 3: Secretion by cell at base line of three cancer cell-lines, and secretion by cell of SW48 at base line, with EGF, and with Cetuximab treatment. The number of biological replicates (n) studied, the mean production of secretome by cell (pg/cell) and the standard deviation are given.

Cell line	n	Mean pg/cell	Standard deviation
MCF7	5	2.2	0.11
SW48	6	4.4	0.06
LIM1215	2	2.5	0.01
SW48 bl	6	4.4	0.06
SW48+EGF	4	2.7	0.29
SW48+Cetux	3	6.6	0.13

3 Conclusions

The package `msmsEDA` provides functions for the exploratory data analysis of label-free LC-MS/MS experiments with SpC to evidence the presence of batch effects or outliers, and also to analyse the residual dispersion of each factor in the experiments to help in deciding the best underlying distribution for inference.

The package `msmsTests` provides functions for inference based on GLMs, with the Poisson distribution, the implementation of the negative binomial distribution in the package `edgeR` [Robinson & Smyth, 2008], or using the quasi-likelihood GLM extension. The model allows for blocking factors to correct for batch effects, and the use of offsets as a flexible and general method of normalization embedded in the model.

Both packages are integrated in the Bioconductor infrastructure and use the `MSnSet` S4 class defined in the package `MSnbase` [Gatto & Lilley, 2012]

4 Methods

4.1 Datasets

The datasets provided in the packages consist of samples of 500 ng of standard yeast digest spiked with different amounts of a mix of 48 equimolar human proteins (UPS1, Sigma-Aldrich). The samples were analyzed using an LTQ Velos-Orbitrap mass spectrometer (Thermo Fisher Scientific, Bremen, Germany) according to the methods detailed elsewhere [Gregori et al., 2012, 2013]. These samples where analysed in batches spanning a few weeks.

Cancer cell lines were cultured in 5% CO2 and 95% humidified atmosphere air at 37°C in Dulbecco's modified Eagle's medium: Nutrient Mixture F-12 (DMEM/F12; Invitrogen, Paisley, UK), supplemented with 10% fetal bovine serum (FBS; Invitrogen), and 2 mM LGlutamine (Invitrogen). Secretomes were prepared as previously described [Villarreal et al., 2013]. Briefly, 4×10^6 cells in exponential phase were seeded in 150 cc tissue culture plates and allowed to grow for 48h. After that, media was aspirated, and cells were washed 5 times, two times with PBS and the last three with serum-free media. After that, cells were maintained in the presence of serum-free media for 24h before collecting the conditioned media (secretome). Within 30 minutes of collecting a secretome, cell number and viability measurements were performed using Countess (Invitrogen, Carlsbad, CA). The cell number calculated by the automatic cell counter was later used to estimate the amount of protein secreted per cell.

4.2 GLMs for inference with SpC

The inference method used in the package `msmsTests` is based on generalized linear models (GLM), which is a natural choice when dealing with counts. A GLM [Agresti, 2002] is specified by three components: (i) The response as a random variable Y and its probability distribution. (ii) A systematic component given by a linear combination of predictor variables. (iii) A link function which relates $E(Y)$ with the linear predictor.

The distribution should belong to the exponential family, which has a probability mass function factorizable as:

$$f(y_i;\theta_i) = a(\theta_i)b(y_i)exp[y_iQ(\theta_i)] \tag{1}$$

Examples are the Poisson distribution, and the negative-binomial when the dispersion parameter ϕ is given. The link function that transforms the mean to the natural parameter θ_i is known as the canonical link. The canonical link for the Poisson or the negative-binomial distribution is the natural logarithm, and the regression model using this link is:

$$log\,\mu_i = \sum_j \beta_j\, x_{ij} \tag{2}$$

where here the x_{ij} are the design matrix elements and β_j the model parameters.

The most basic distribution when modelling counts is the Poisson distribution, a distribution with just one parameter, μ, the mean.

$$Pr(X = k) = \frac{\mu^k e^{-\mu}}{k!} \tag{3}$$

This distribution explains the uncertainty in the number of SpC positively identified as belonging to a given protein, when the expected number of SpC for this protein at a given concentration is μ. The Poisson distribution has the property that the variance equals the mean. The higher the number of expected counts, the higher is its variance. When the only source of variation comes from the sampling process, as when running technical replicates, the Poisson distribution works very well. Nevertheless when doing biological experiments, to the typical variation in sampling technical replicates we have to add the biological variability expected of individuals belonging to the same biological condition. In these circumstances the Poisson model will underestimate the variance and the inference may bring to false positives in differential expression. This phenomenon is known as overdispersion. [Agresti, 2002] The immediate alternative is the negative-binomial (NB) distribution, which allows for overdispersion. The probability mass function of a NB random variable with mean μ and dispersion ϕ [Agresti, 2002; Robinson & Smyth, 2008] is given by:

$$Pr(X = k) = \frac{\Gamma(k + \phi^{-1})}{\Gamma(\phi^{-1})\,\Gamma(k+1)} \left(\frac{1}{1+\mu\phi}\right)^{\phi^{-1}} \left(\frac{\mu}{\phi^{-1}+\mu}\right)^{k} \tag{4}$$

where the variance is a function of both mean and dispersion.

$$Var(X) = \mu + \phi\mu^2 \tag{5}$$

When $\phi \to 0$ the NB reduces to the Poisson distribution. Values of ϕ greater than 0 bring to overdispersed distributions. Strictly speaking any value $\phi > -\mu^{-1}$ is permitted by the model, allowing for some degree of subdispersion.

An extension of the GLM, making abstraction of the true distribution, considers the mean-variance relationship

$$Var(Y) = \psi\mu_i \tag{6}$$

for some constant ψ. The case $\psi > 1$ corresponds to overdispersion. The likelihood equations for this model are identical to the Poisson model, and the model parameter estimates are the same, but ψ is not assumed to be fixed at 1 and estimated from the data. This brings to the quasi-likelihood model which fits the Poisson model and multiplies the standard error estimates of the model parameters by the square root of $\hat{\psi}$, thus adjusting inference for overdispersion. [Agresti, 2002]

The p-value for differential secretion is obtained from the log likelihood ratio comparing the model given by 11 with the model with $\beta = 0$. Multitest p-value adjustment with FDR control is done by the Benjamini-Hochberg method [Benjamini & Hochberg, 1995].

4.3 Sample size normalization

The most extended method of normalization assumes that for a given amount of total protein, the sum of SpC should be the same. In general when GLM models are used [Choi et al., 2008; Li et al., 2010; Leitch et al., 2012] the normalization is implicitly done with the help of an offset term in the model [Agresti, 2002].

$$E[y] = \mu$$

$$log\left(\frac{\mu}{size}\right) = \alpha + \beta x$$

$$log(\mu) = log(size) + \alpha + \beta x \tag{7}$$

where μ is the expected expression of a given protein, $size$ is the normalizing condition, and α and β are the model parameters, with x equal to 0 for the control condition, or equal to one for the tumor condition. The term $log(size)$ is the *offset*. Different covariates or blocking factors may be added to this simple model. This is independent of the underlying distribution for the expression level y of this protein.

The underlying assumption is that the measured substance has been produced by equal number of cells in either state. This assumption could be accepted when analysing plasma or interstitial fluid. But it is more questionable when analysing cell-line secretomes.

4.4 Cell-to-cell normalization

We intend to compare the proteins secreted by one single cell in each of two given biological states of a cell line, even when one state is globally stimulated or depressed with respect to the other. Suppose we gather $Q_j\,\mu$g of total protein secreted by n_j cells in the j-th biological condition, of which $q\,\mu$g are digested and injected into the LC-MS/MS system to be measured. The total quantity of digested protein measured is the same for

the two conditions. The ratio q/Q_j gives the proportion of total secreted protein that gets measured for condition j, hence $(q/Q_j)n_j$ is the number of cells which secreted the $q\,\mu$g in the j-th biological condition. Then we obtain the number of cells which produced the $q\,\mu$g in the j-th condition as in equation 8.

$$c_j = \frac{q}{Q_j} n_j \tag{8}$$

On the other hand given the expected SpC value μ of a protein at a given concentration in the total $q\,\mu$g, the expected value per cell is given by equation 9

$$E\left[\frac{y}{c_j}\right] = \frac{\mu}{c_j} \tag{9}$$

As the total protein measured for the two biological conditions is the same, the factor q contributes equally to both conditions and may be removed, bringing to equation 10 where the mass scale is undefined but equal for both conditions.

$$E\left[\frac{y}{c_j}\right] = \frac{\mu}{n_j/Q_j} \tag{10}$$

This allows to formulate a GLM model, as in equation 7, taking into account the total spectral counts by sample, $size$, the protein production rate of each condition, Q/n, the treatment factor, X, and a blocking factor to account for non controlled factors leading to batch effects, Z, as in equation 11.

$$log(\mu) = log\left(\frac{n}{Q}\right) + log(size) + \alpha + \beta x + \gamma z \tag{11}$$

With this model the effect size, as logFC, is estimated as in equation 12.

$$FC_z = \frac{\mu_A/(size_A n_A/Q_A)}{\mu_B/(size_B n_B/Q_B)} = \frac{exp(\alpha+\beta+\gamma z)}{exp(\alpha+\gamma z)} = exp(\beta)$$

$$\widehat{logFC} = log_2(exp(\hat{\beta})) = 1.44\hat{\beta} \tag{12}$$

where the subindex A stands for treatment, with $x = 1$, and subindex B for control, with $x = 0$.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

JG developed the packages and authored the paper. JV supervised the experiments and provided expertise in protemomics. AS supervised the statistic work and the development of the packages.

Acknowledgements

We are indebted to Laura Villarreal, Olga Méndez, Theodora Katsila and Candida Salvans for their collaboration in the experimental data that has been used in this manuscript.

References

Agresti, A. (2002). *Categorical Data Analysis*. Wiley Series in Probability and Statistics - Applied Probability and Statistics Section Series. Wiley.

Benjamini, Y. & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *57*(1), 289–300.

Choi, H., Fermin, D., & Nesvizhskii, A. I. (2008). Significance analysis of spectral count data in label-free shotgun proteomics. *7*(12), 23732385.

Gatto, L. & Lilley, K. S. (2012). MSnbase-an R/Bioconductor package for isobaric tagged mass spectrometry data visualization, processing and quantitation. *Bioinformatics*, *28*(2), 288–289. PMID: 22113085.

Gentleman, R. C., Carey, V. J., Bates, D. M., & others (2004). Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, *5*, R80.

Gregori, J., Villarreal, L., Méndez, O., Sánchez, A., Baselga, J., & Villanueva, J. (2012). Batch effects correction improves the sensitivity of significance tests in spectral counting-based comparative discovery proteomics. *75*(13), 3938–3951. PMID: 22588121.

Gregori, J., Villarreal, L., Sánchez, A., Baselga, J., & Villanueva, J. (2013). An effect size filter improves the reproducibility in spectral counting-based comparative proteomics. *95*, 55–65. PMID: 23770383.

Gronborg, M., Kristiansen, T. Z., Iwahori, A., Chang, R., Reddy, R., Sato, N., Molina, H., Jensen, O. N., Hruban, R. H., Goggins, M. G., Maitra, A., & Pandey, A. (2006). Biomarker discovery from pancreatic cancer secretome using a differential proteomic approach. *Mol. Cell Proteomics*, *5*(1), 157–171. PMID: 16215274.

Gygi, S. P., Rochon, Y., Franza, B. R., & Aebersold, R. (1999). Correlation between protein and mRNA abundance in yeast. *Mol. Cell. Biol.*, *19*(3), 1720–1730. PMID: 10022859.

Knudsen, S. *Cancer Diagnostics with DNA Microarrays*. Hoboken, NJ: John Wiley and Sons.

Lawlor, K., Nazarian, A., Lacomis, L., Tempst, P., & Villanueva, J. (2009). Pathway-based biomarker search by high-throughput proteomics profiling of secretomes. *8*(3), 1489–1503.

Lazar, C., Meganck, S., Taminau, J., Steenhoff, D., Coletta, A., Molter, C., Weiss-Solís, D. Y., Duque, R., Bersini, H., & Nowé, A. (2013). Batch effect removal methods for microarray gene expression data integration: a survey. *Brief. Bioinformatics*, *14*(4), 469–490. PMID: 22851511.

Leitch, M. C., Mitra, I., & Sadygov, R. G. (2012). Generalized linear and mixed models for label-free shotgun proteomics. *Stat Interface*, *5*(1), 89–98. PMID: 22822415.

Li, M., Gray, W., Zhang, H., Chung, C. H., Billheimer, D., Yarbrough, W. G., Liebler, D. C., Shyr, Y., & Slebos, R. J. C. (2010). Comparative shotgun proteomics using spectral count data and quasi-likelihood modeling. *9*(8), 4295–4305.

Lundgren, D. H., Hwang, S.-I., Wu, L., & Han, D. K. (2010). Role of spectral counting in quantitative proteomics. *Expert Rev Proteomics*, *7*(1), 39–53. PMID: 20121475.

Luo, J., Schumacher, M., Scherer, A., Sanoudou, D., Megherbi, D., Davison, T., Shi, T., Tong, W., Shi, L., Hong, H., Zhao, C., Elloumi, F., Shi, W., Thomas, R., Lin, S., Tillinghast, G., Liu, G., Zhou, Y., Herman, D., Li, Y., Deng, Y., Fang, H., Bushel, P., Woods, M., & Zhang, J. (2010). A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data. *10*(4), 278–291.

Mathias, R. A., Wang, B., Ji, H., Kapp, E. A., Moritz, R. L., Zhu, H.-J., & Simpson, R. J. (2009). Secretome-based proteomic profiling of ras-transformed MDCK cells reveals extracellular modulators of epithelial-mesenchymal transition. *J. Proteome Res.*, *8*(6), 2827–2837. PMID: 19296674.

Neilson, K. A., Ali, N. A., Muralidharan, S., Mirzaei, M., Mariani, M., Assadourian, G., Lee, A., van Sluyter, S. C., & Haynes, P. A. (2011). Less label, more free: Approaches in label-free quantitative mass spectrometry. *11*(4), 535–553.

Patel, V. J., Thalassinos, K., Slade, S. E., Connolly, J. B., Crombie, A., Murrell, J. C., & Scrivens, J. H. (2009). A comparison of labeling and label-free mass spectrometry-based proteomics approaches. *J. Proteome Res.*, *8*(7), 3752–3759. PMID: 19435289.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Rifai, N., Gillette, M. A., & Carr, S. A. (2006). Protein biomarker discovery and validation: the long and uncertain path to clinical utility. *Nat. Biotechnol.*, *24*(8), 971–983. PMID: 16900146.

Robinson, M. D. & Smyth, G. K. (2008). Small-sample estimation of negative binomial dispersion, with applications to sage data. *Biostatistics*, *9*(2), 321–332.

Sandin, M., Krogh, M., Hansson, K., & Levander, F. (2011). Generic workflow for quality assessment of quantitative label-free LC-MS analysis. *Proteomics*, *11*(6), 1114–1124. PMID: 21298787.

Scherer, A. (2009). *Batch Effects and Noise in Microarray Experiments: Sources and Solutions*. Wiley Series in Probability and Statistics. Wiley.

Shi, L., et al., & MAQC-Consortium (2006). The MicroArray quality control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nat Biotech*, *24*(9), 1151–1161.

Shi, L., et al., & MAQC-Consortium (2008). The balance of reproducibility, sensitivity, and specificity of lists of differentially expressed genes in microarray studies. *9 Suppl 9*, S10. PMID: 18793455.

Stastna, M. & Van Eyk, J. E. (2012). Secreted proteins as a fundamental source for biomarker discovery. *Proteomics*, *12*(4-5), 722–735. PMID: 22247067.

Tirumalai, R. S., Chan, K. C., Prieto, D. A., Issaq, H. J., Conrads, T. P., & Veenstra, T. D. (2003). Characterization of the low molecular weight human serum proteome. *Molecular & Cellular Proteomics*, *2*(10), 1096–1103.

Villarreal, L., Méndez, O., Salvans, C., Gregori, J., Baselga, J., & Villanueva, J. (2013). Unconventional secretion is a major contributor of cancer cell line secretomes. *12*(5), 1046–1060. PMID: 23268930.

Zhu, W., Smith, J. W., & Huang, C.-M. (2010). Mass spectrometry-based label-free quantitative proteomics. *J. Biomed. Biotechnol.*, *2010*, 840–518. PMID: 19911078.

A Appendix with R scripts

A.1 Script to obtain the results in table 2

```
library(msmsEDA)
library(msmsTests)

data(msms.dataset)
msms.dataset
dim(msms.dataset)
head(pData(msms.dataset))
table(pData(msms.dataset)$treat)
table(pData(msms.dataset)$batch)

###  Expression matrix and factors
msms.spc <- exprs(msms.dataset)
dim(msms.spc)
treat <- pData(msms.dataset)[,1]
batch <- pData(msms.dataset)[,2]
table(treat,batch)

###  Subset to batch '0302'
e <- msms.dataset[,batch=="0302"]
msms.spc <- exprs(e)
treat <- pData(e)[,1]
### Preprocess: Remove all zero rows
e <- pp.msms.data(e)
dim(e)

###  TP indices
idx <- grep("HUMAN",rownames(msms.spc))

###  Summary of mean Spc by treatment level
mSpC <- t( apply(msms.spc[idx,],1,function(x) tapply(x,treat,mean)) )
apply(mSpC,2,summary)


#########################################################################
###                    P O I S S O N

### Normalizing condition by total SpC
div <- apply(exprs(e),2,sum)

### Models: Null, treatment, treatment+batch
null1.f <- "y~1"
alt1.f <- "y~treat"

### Poisson GLM. Null1 vs Alternative 1
pois1.res <- msms.glm.pois(e,alt1.f,null1.f,div=div)
str(pois1.res)

### DEPs on unadjusted p-values
sum(pois1.res$p.value<=0.01)
### DEPs on multitest adjusted p-values
adjp <- p.adjust(pois1.res$p.value,method="BH")
```

```
sum(adjp<=0.01)

### The top features
o <- order(pois1.res$p.value)
head(pois1.res[o,],20)

### How the UPS1 proteins get ordered in the list
grep("HUMAN",rownames(pois1.res[o,]))

###  Truth table at 0.01
nh <- length(grep("HUMAN",featureNames(e)))
ny <- length(grep("HUMAN",featureNames(e),invert=TRUE))
tp <- length(grep("HUMAN",rownames(pois1.res)[adjp<=0.01]))
fp <- sum(adjp<=0.01)-tp
(tt.pois1 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))
###  Truth table at 0.05
tp <- length(grep("HUMAN",rownames(pois1.res)[adjp<=0.05]))
fp <- sum(adjp<=0.05)-tp
(tt.pois12 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

###  Cut-off values for a relevant protein as biomarker
alpha.cut <- 0.05
SpC.cut <- 2
lFC.cut <- 0.8

###  Reproducible results
pois1.tbl <- test.results(pois1.res,e,pData(e)$treat,"U600","U200",div,
                          alpha=alpha.cut,minSpC=SpC.cut,minLFC=lFC.cut,
                          method="BH")$tres
(pois1.nms <- rownames(pois1.tbl)[pois1.tbl$DEP])

###  Truth table
ridx <- grep("HUMAN",pois1.nms)
tp <- length(ridx)
fp <- length(pois1.nms)-length(ridx)
(tt.pois2 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

tt.pois <- rbind(tt.pois12,tt.pois1,tt.pois2)
rownames(tt.pois) <- c("Alt1 - 0.05","Alt1 - 0.01","Alt1+filter")
tt.pois


##############################################################################
###                 Q U A S I - L I K E L I H O O D

### QLLL GLM. Null1 vs Alternative 1
qlll1.res <- msms.glm.qlll(e,alt1.f,null1.f,div=div)
str(qlll1.res)

### DEPs on unadjusted p-values
sum(qlll1.res$p.value<=0.01)
### DEPs on multitest adjusted p-values
adjp <- p.adjust(qlll1.res$p.value,method="BH")
sum(adjp<=0.01)
```

```
### The top features
o <- order(qlll1.res$p.value)
head(qlll1.res[o,],20)

### How the UPS1 proteins get ordered in the list
grep("HUMAN",rownames(qlll1.res[o,]))

###  Truth table at 0.01
nh <- length(grep("HUMAN",featureNames(e)))
ny <- length(grep("HUMAN",featureNames(e),invert=TRUE))
tp <- length(grep("HUMAN",rownames(qlll1.res)[adjp<=0.01]))
fp <- sum(adjp<=0.01)-tp
(tt.qlll1 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))
###  Truth table at 0.05
tp <- length(grep("HUMAN",rownames(qlll1.res)[adjp<=0.05]))
fp <- sum(adjp<=0.05)-tp
(tt.qlll12 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))


###  Reproducible results
qlll1.tbl <- test.results(qlll1.res,e,pData(e)$treat,"U600","U200",div,
                          alpha=alpha.cut,minSpC=SpC.cut,minLFC=lFC.cut,
                          method="BH")$tres
(qlll1.nms <- rownames(qlll1.tbl)[qlll1.tbl$DEP])

###  Truth table
ridx <- grep("HUMAN",qlll1.nms)
tp <- length(ridx)
fp <- length(qlll1.nms)-length(ridx)
(tt.qlll2 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))


tt.qlll <- rbind(tt.qlll12,tt.qlll1,tt.qlll2)
rownames(tt.qlll) <- c("Alt1 - 0.05","Alt1 - 0.01","Alt1+filter")
tt.qlll


#####################################################################
###                NEGATIVE BINOMIAL

### Negative binomial GLM. Null1 vs Alternative 1
nb1.res <- msms.edgeR(e,alt1.f,null1.f,div=div,fnm="treat")
str(nb1.res)

### DEPs on unadjusted p-values
sum(nb1.res$p.value<=0.01)
### DEPs on multitest adjusted p-values
adjp <- p.adjust(nb1.res$p.value,method="BH")
sum(adjp<=0.01)

### The top features
o <- order(nb1.res$p.value)
head(nb1.res[o,],20)

### How the UPS1 proteins get ordered in the list
```

```
grep("HUMAN",rownames(nb1.res[o,]))

###  Truth table at 0.01
nh <- length(grep("HUMAN",featureNames(e)))
ny <- length(grep("HUMAN",featureNames(e),invert=TRUE))
tp <- length(grep("HUMAN",rownames(nb1.res)[adjp<=0.01]))
fp <- sum(adjp<=0.01)-tp
(tt.nb1 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))
###  Truth table at 0.05
tp <- length(grep("HUMAN",rownames(nb1.res)[adjp<=0.05]))
fp <- sum(adjp<=0.05)-tp
(tt.nb12 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))


###  Reproducible results
nb1.tbl <- test.results(nb1.res,e,pData(e)$treat,"U600","U200",div,
                        alpha=alpha.cut,minSpC=SpC.cut,minLFC=lFC.cut,
                        method="BH")$tres
(nb1.nms <- rownames(nb1.tbl)[nb1.tbl$DEP])

###  Truth table
ridx <- grep("HUMAN",nb1.nms)
tp <- length(ridx)
fp <- length(nb1.nms)-length(ridx)
(tt.nb2 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

tt.nb <- rbind(tt.nb12,tt.nb1,tt.nb2)
rownames(tt.nb) <- c("Alt1 - 0.05","Alt1 - 0.01","Alt1+filter")
tt.nb


#############################################################################
###                 G L O B A L   R E S U L T S

n <- nrow(tt.pois*3)
df <- data.frame(Model=character(n),Distr=character(n),
                 TP=integer(n),FP=integer(n),TN=integer(n),FN=integer(n),
                 stringsAsFactors=FALSE)
ops <- rownames(tt.pois)
for(i in 1:length(ops))
{ k <- (i*3-2):(i*3)
  df[k,1] <- ops[i]
  df[k,2] <- c("Pois","QL","NB")
  df[k[1],3:6] <- tt.pois[i,]
  df[k[2],3:6] <- tt.qlll[i,]
  df[k[3],3:6] <- tt.nb[i,]
}
df

save(tt.pois,tt.qlll,tt.nb,df,file="OneBatchTruthTables.RData")

tpr <- df$TP/(df$TP+df$FN)
tpr <- round(tpr*100,2)
acc <- (df$TP+df$TN)/(df$TP+df$FN+df$FP+df$TN)
acc <- round(acc*100,2)
```

```
ppv <- df$TP/(df$TP+df$FP)
ppv <- round(ppv*100,2)
fdr <- df$FP/(df$TP+df$FP)
fdr <- round(fdr*100,2)
div <- sqrt((df$TP+df$FP)*(df$TP+df$FN)*(df$TN+df$FP)*(df$TN+df$FN))
mcc <- (df$TP*df$TN-df$FP*df$FN)/div

df2 <- data.frame(df,TPR=tpr,ACC=acc,FDR=fdr,PPV=ppv)
df2

save(df2,tt.pois,tt.qlll,tt.nb,df,file="OneBatchFullTruthTables.RData")
```

A.2 Script to obtain figure 1 and the results in table 1

```
library(msmsEDA)
library(msmsTests)

data(msms.dataset)
msms.dataset
dim(msms.dataset)
head(pData(msms.dataset))
table(pData(msms.dataset)$treat)
table(pData(msms.dataset)$batch)

###  Expression matrix and factors
msms.spc <- exprs(msms.dataset)
dim(msms.spc)
treat <- pData(msms.dataset)[,1]
batch <- pData(msms.dataset)[,2]
table(treat,batch)

###  TP indices
idx <- grep("HUMAN",rownames(msms.spc))

###  Summary of mean Spc by treatment level
mSpC <- t( apply(msms.spc[idx,],1,function(x) tapply(x,treat,mean)) )
apply(mSpC,2,summary)

###  Summary of mean Spc by batch level
mSpCb <- t( apply(msms.spc[idx,],1,function(x) tapply(x,batch,mean)) )
apply(mSpCb,2,summary)

### Preprocess: Remove all zero rows
e <- pp.msms.data(msms.dataset)
dim(e)

### EXPLORATORY DATA ANALYSIS

### Principal Components Analysis
pdf(file="paper.PCA.pdf",paper="a4",width=6,height=5)
counts.pca(e,facs=batch,snms=as.character(treat))
dev.off()
###  Hierarchical clustering
```

```
pdf(file="paper.HC.pdf",paper="a4",width=5,height=6)
counts.hc(e,facs=treat)
dev.off()
###  Heatmap
pdf(file="paper.Heatmap.pdf",paper="a4",width=6,height=8)
counts.heatmap(e,fac=treat)
dev.off()

### Normalizing condition by total SpC
div <- apply(exprs(e),2,sum)

### Models: Null, treatment, treatment+batch
null1.f <- "y~1"
alt1.f <- "y~treat"
null2.f <- "y~batch"
alt2.f <- "y~treat+batch"


##############################################################################
###                        P O I S S O N

### Poisson GLM. Null1 vs Alternative 1
pois1.res <- msms.glm.pois(e,alt1.f,null1.f,div=div)
str(pois1.res)

### DEPs on unadjusted p-values
sum(pois1.res$p.value<=0.01)
### DEPs on multitest adjusted p-values
adjp <- p.adjust(pois1.res$p.value,method="BH")
sum(adjp<=0.01)

### The top features
o <- order(pois1.res$p.value)
head(pois1.res[o,],20)

### How the UPS1 proteins get ordered in the list
grep("HUMAN",rownames(pois1.res[o,]))

###  Truth table at 0.01
nh <- length(grep("HUMAN",featureNames(e)))
ny <- length(grep("HUMAN",featureNames(e),invert=TRUE))
tp <- length(grep("HUMAN",rownames(pois1.res)[adjp<=0.01]))
fp <- sum(adjp<=0.01)-tp
(tt.pois1 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))
###  Truth table at 0.05
tp <- length(grep("HUMAN",rownames(pois1.res)[adjp<=0.05]))
fp <- sum(adjp<=0.05)-tp
(tt.pois12 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))


###  Cut-off values for a relevant protein as biomarker
alpha.cut <- 0.05
SpC.cut <- 2
lFC.cut <- 0.8
```

```
###  Reproducible results
pois1.tbl <- test.results(pois1.res,e,pData(e)$treat,"U600","U200",div,
                          alpha=alpha.cut,minSpC=SpC.cut,minLFC=lFC.cut,
                          method="BH")$tres
(pois1.nms <- rownames(pois1.tbl)[pois1.tbl$DEP])

###  Truth table
ridx <- grep("HUMAN",pois1.nms)
tp <- length(ridx)
fp <- length(pois1.nms)-length(ridx)
(tt.pois2 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))


### Poisson GLM. Null2 vs Alternative 2
pois2.res <- msms.glm.pois(e,alt2.f,null2.f,div=div)
str(pois2.res)

### DEPs on unadjusted p-values
sum(pois2.res$p.value<=0.01)
### DEPs on multitest adjusted p-values
adjp <- p.adjust(pois2.res$p.value,method="BH")
sum(adjp<=0.01)

### The top features
o <- order(pois2.res$p.value)
head(pois2.res[o,],20)

### How the UPS1 proteins get ordered in the list
grep("HUMAN",rownames(pois2.res[o,]))

###  Truth table at 0.01
nh <- length(grep("HUMAN",featureNames(e)))
ny <- length(grep("HUMAN",featureNames(e),invert=TRUE))
tp <- length(grep("HUMAN",rownames(pois2.res)[adjp<=0.01]))
fp <- sum(adjp<=0.01)-tp
(tt.pois3 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

###  Truth table at 0.05
tp <- length(grep("HUMAN",rownames(pois2.res)[adjp<=0.05]))
fp <- sum(adjp<=0.05)-tp
(tt.pois32 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))


###  Reproducible results
pois2.tbl <- test.results(pois2.res,e,pData(e)$treat,"U600","U200",div,
                          alpha=alpha.cut,minSpC=SpC.cut,minLFC=lFC.cut,
                          method="BH")$tres
(pois2.nms <- rownames(pois2.tbl)[pois2.tbl$DEP])

###  Truth table
ridx <- grep("HUMAN",pois2.nms)
tp <- length(ridx)
fp <- length(pois2.nms)-length(ridx)
(tt.pois4 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))
```

```
tt.pois <- rbind(tt.pois12,tt.pois1,tt.pois2,tt.pois32,tt.pois3,tt.pois4)
rownames(tt.pois) <- c("Alt1 - 0.05","Alt1 - 0.01","Alt1+filter",
                       "Alt2 - 0.05","Alt2 - 0.01","Alt2+filter")
tt.pois


#########################################################################
###                 QUASI-LIKELIHOOD

### QLLL GLM. Null1 vs Alternative 1
qlll1.res <- msms.glm.qlll(e,alt1.f,null1.f,div=div)
str(qlll1.res)

### DEPs on unadjusted p-values
sum(qlll1.res$p.value<=0.01)
### DEPs on multitest adjusted p-values
adjp <- p.adjust(qlll1.res$p.value,method="BH")
sum(adjp<=0.01)

### The top features
o <- order(qlll1.res$p.value)
head(qlll1.res[o,],20)

### How the UPS1 proteins get ordered in the list
grep("HUMAN",rownames(qlll1.res[o,]))

###  Truth table at 0.01
nh <- length(grep("HUMAN",featureNames(e)))
ny <- length(grep("HUMAN",featureNames(e),invert=TRUE))
tp <- length(grep("HUMAN",rownames(qlll1.res)[adjp<=0.01]))
fp <- sum(adjp<=0.01)-tp
(tt.qlll1 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))
###  Truth table at 0.05
tp <- length(grep("HUMAN",rownames(qlll1.res)[adjp<=0.05]))
fp <- sum(adjp<=0.05)-tp
(tt.qlll12 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))


###  Reproducible results
qlll1.tbl <- test.results(qlll1.res,e,pData(e)$treat,"U600","U200",div,
                          alpha=alpha.cut,minSpC=SpC.cut,minLFC=lFC.cut,
                          method="BH")$tres
(qlll1.nms <- rownames(qlll1.tbl)[qlll1.tbl$DEP])

###  Truth table
ridx <- grep("HUMAN",qlll1.nms)
tp <- length(ridx)
fp <- length(qlll1.nms)-length(ridx)
(tt.qlll2 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))


### QLLL GLM. Null2 vs Alternative 2
qlll2.res <- msms.glm.qlll(e,alt2.f,null2.f,div=div)
str(qlll2.res)
```

```
### DEPs on unadjusted p-values
sum(qlll2.res$p.value<=0.01)
### DEPs on multitest adjusted p-values
adjp <- p.adjust(qlll2.res$p.value,method="BH")
sum(adjp<=0.01)

### The top features
o <- order(qlll2.res$p.value)
head(qlll2.res[o,],20)

### How the UPS1 proteins get ordered in the list
grep("HUMAN",rownames(qlll2.res[o,]))

###  Truth table at 0.01
nh <- length(grep("HUMAN",featureNames(e)))
ny <- length(grep("HUMAN",featureNames(e),invert=TRUE))
tp <- length(grep("HUMAN",rownames(qlll2.res)[adjp<=0.01]))
fp <- sum(adjp<=0.01)-tp
(tt.qlll3 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))
###  Truth table at 0.05
tp <- length(grep("HUMAN",rownames(qlll2.res)[adjp<=0.05]))
fp <- sum(adjp<=0.05)-tp
(tt.qlll32 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

###  Reproducible results
qlll2.tbl <- test.results(qlll2.res,e,pData(e)$treat,"U600","U200",div,
                          alpha=alpha.cut,minSpC=SpC.cut,minLFC=lFC.cut,
                          method="BH")$tres
(qlll2.nms <- rownames(qlll2.tbl)[qlll2.tbl$DEP])

###  Truth table
ridx <- grep("HUMAN",qlll2.nms)
tp <- length(ridx)
fp <- length(qlll2.nms)-length(ridx)
(tt.qlll4 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

tt.qlll <- rbind(tt.qlll12,tt.qllll,tt.qlll2,tt.qlll32,tt.qlll3,tt.qlll4)
rownames(tt.qlll) <- c("Alt1 - 0.05","Alt1 - 0.01","Alt1+filter",
                       "Alt2 - 0.05","Alt2 - 0.01","Alt2+filter")
tt.qlll


##############################################################################
###                NEGATIVE BINOMIAL

### Negative binomial GLM. Null1 vs Alternative 1
nb1.res <- msms.edgeR(e,alt1.f,null1.f,div=div,fnm="treat")
str(nb1.res)

### DEPs on unadjusted p-values
sum(nb1.res$p.value<=0.01)
### DEPs on multitest adjusted p-values
adjp <- p.adjust(nb1.res$p.value,method="BH")
sum(adjp<=0.01)
```

```
### The top features
o <- order(nb1.res$p.value)
head(nb1.res[o,],20)

### How the UPS1 proteins get ordered in the list
grep("HUMAN",rownames(nb1.res[o,]))

###  Truth table at 0.01
nh <- length(grep("HUMAN",featureNames(e)))
ny <- length(grep("HUMAN",featureNames(e),invert=TRUE))
tp <- length(grep("HUMAN",rownames(nb1.res)[adjp<=0.01]))
fp <- sum(adjp<=0.01)-tp
(tt.nb1 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))
###  Truth table at 0.05
tp <- length(grep("HUMAN",rownames(nb1.res)[adjp<=0.05]))
fp <- sum(adjp<=0.05)-tp
(tt.nb12 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))


###  Reproducible results
nb1.tbl <- test.results(nb1.res,e,pData(e)$treat,"U600","U200",div,
                        alpha=alpha.cut,minSpC=SpC.cut,minLFC=lFC.cut,
                        method="BH")$tres
(nb1.nms <- rownames(nb1.tbl)[nb1.tbl$DEP])

###  Truth table
ridx <- grep("HUMAN",nb1.nms)
tp <- length(ridx)
fp <- length(nb1.nms)-length(ridx)
(tt.nb2 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))


### Negative binomial GLM. Null2 vs Alternative 2
nb2.res <- msms.edgeR(e,alt2.f,null2.f,div=div,fnm="treat")
str(nb2.res)

### DEPs on unadjusted p-values
sum(nb2.res$p.value<=0.01)
### DEPs on multitest adjusted p-values
adjp <- p.adjust(nb2.res$p.value,method="BH")
sum(adjp<=0.01)

### The top features
o <- order(nb2.res$p.value)
head(nb2.res[o,],20)

### How the UPS1 proteins get ordered in the list
grep("HUMAN",rownames(nb2.res[o,]))

###  Truth table at 0.01
nh <- length(grep("HUMAN",featureNames(e)))
ny <- length(grep("HUMAN",featureNames(e),invert=TRUE))
tp <- length(grep("HUMAN",rownames(nb2.res)[adjp<=0.01]))
fp <- sum(adjp<=0.01)-tp
(tt.nb3 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))
```

```
###  Truth table at 0.05
tp <- length(grep("HUMAN",rownames(nb2.res)[adjp<=0.05]))
fp <- sum(adjp<=0.05)-tp
(tt.nb32 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

###  Reproducible results
nb2.tbl <- test.results(nb2.res,e,pData(e)$treat,"U600","U200",div,
                        alpha=alpha.cut,minSpC=SpC.cut,minLFC=lFC.cut,
                        method="BH")$tres
(nb2.nms <- rownames(nb2.tbl)[nb2.tbl$DEP])

###  Truth table
ridx <- grep("HUMAN",nb2.nms)
tp <- length(ridx)
fp <- length(nb2.nms)-length(ridx)
(tt.nb4 <- data.frame(TP=tp,FP=fp,TN=ny-fp,FN=nh-tp))

tt.nb <- rbind(tt.nb12,tt.nb1,tt.nb2,tt.nb32,tt.nb3,tt.nb4)
rownames(tt.nb) <- c("Alt1 - 0.05","Alt1 - 0.01","Alt1+filter",
                     "Alt2 - 0.05","Alt2 - 0.01","Alt2+filter")
tt.nb


#####################################################################
###                 G L O B A L   R E S U L T S

n <- nrow(tt.pois*3)
df <- data.frame(Model=character(n),Distr=character(n),
                 TP=integer(n),FP=integer(n),TN=integer(n),FN=integer(n),
                 stringsAsFactors=FALSE)
ops <- rownames(tt.pois)
for(i in 1:length(ops))
{ k <- (i*3-2):(i*3)
  df[k,1] <- ops[i]
  df[k,2] <- c("Pois","QL","NB")
  df[k[1],3:6] <- tt.pois[i,]
  df[k[2],3:6] <- tt.qlll[i,]
  df[k[3],3:6] <- tt.nb[i,]
}
df

save(tt.pois,tt.qlll,tt.nb,df,file="TruthTables.RData")

tpr <- df$TP/(df$TP+df$FN)
tpr <- round(tpr*100,2)
acc <- (df$TP+df$TN)/(df$TP+df$FN+df$FP+df$TN)
acc <- round(acc*100,2)
ppv <- df$TP/(df$TP+df$FP)
ppv <- round(ppv*100,2)
fdr <- df$FP/(df$TP+df$FP)
fdr <- round(fdr*100,2)
div <- sqrt((df$TP+df$FP)*(df$TP+df$FN)*(df$TN+df$FP)*(df$TN+df$FN))
mcc <- (df$TP*df$TN-df$FP*df$FN)/div

df2 <- data.frame(df,TPR=tpr,ACC=acc,FDR=fdr,PPV=ppv)
```

```
df2

save(df2, tt.pois, tt.qlll, tt.nb, df, file="FullTruthTables.RData")
```